Deep Belief Networks

4-27-16
Announcement

We are extending the deadline for the final project.

You now have an extra week (due Monday 5/9).

I still strongly recommend planning to finish your implementation this week. This will give you time to run experiments, make tweaks, and try alternatives. You should also plan to finish your experiments a few days before the deadline so that you have time to write the paper.
Recall: sigmoid activation function

\[
thresh(x) = \begin{cases} 
0 & x < 0 \\
1 & x \geq 0
\end{cases}
\]

\[
\sigma(x) = \frac{1}{1 + e^{-x}}
\]
The problem with sigmoids: vanishing gradient

\[ w_i = w_i + \eta \delta x_i \]

\[ \delta_o = o_o(1 - o_o)(t_o - o_o) \]

\[ \delta_h = o_h(1 - o_h) \left( \sum_k \delta_k w_{hk} \right) \]

The \( \delta \) terms tend to be small, and get smaller at each layer, meaning that backpropagation takes a very long time to update weights.
First key idea: unsupervised weight initialization

One layer at a time, train the weights to encode a compressed representation of the previous layer.

This can be done with:

- Restricted Boltzman machines
- Auto-encoders

After this pre-training phase (which uses only unlabeled data), train the network on labeled data with standard backprop.
Second key idea: ditch the sigmoids

How did we end up using sigmoid activation functions?

- Biological neurons act kind of like threshold functions.
- Sigmoids give continuous, differentiable approximations to thresholds.

But maybe for artificial neural networks, there are better activation functions.
Rectifier linear units

\[
ReLU(x) = \begin{cases} 
0 & x < 0 \\
 x & x \geq 0 
\end{cases}
\]

\[
softmax(x) = \ln(1 + e^x)
\]
Advantages of ReLUs

- The gradient never vanishes.
- Computing $\delta$ is trivial.
  - The derivative is 1 or 0 everywhere.
- Activations tend to be sparse (lots of zeros).
- Unbounded range $\Rightarrow$ better for regression.
Third key idea: get TONS of data

- Make the training fast with GPU parallelization.
- Instead of presenting the same examples many times, go get more data.

Example data sets for deep learning:

- Millions of youtube videos.
- Tens of thousands of go games.